

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

---

Application No.:	10/074,092	§	Examiner:	Duong, Oanh L.
Filed:	February 12, 2002	§	Group/Art Unit:	2155
Inventors:		§	Atty. Dkt. No:	5681-05700
Darpan Dinker		§		P7010
Mahesh Kannan		§		
Pramod Gopinath		§		
		§		
Title:	Distributed Data System	§		
	with Process Co-Location	§		
	and Out-of-Process	§		
	Communication.	§		
		§		

---

**REPLY BRIEF**

**Mail Stop Appeal Brief - Patents**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Dear Sir:

This brief is in reply to the Examiner's Answered dated December 6, 2006. Appellants respectfully request that this Reply Brief be entered pursuant to 37 C.F.R. § 41.41 and considered by the Board of Patent Appeals and Interferences.

## **REPLY TO EXAMINER'S ANSWER**

### **First Ground of Rejection:**

#### **Claims 1 - 5, 7-9, 13 – 20, 21-24, and 28-33:**

Appellants have argued that the rejection of claim 1 is improper because the art relied upon by the Examiner explicitly does not teach a distributed data system comprising a plurality of nodes including at least one in-process and at least one out-of-process node, as recited in claim 1. Appellants have pointed out that the AAPA, relied upon by the Examiner, clearly states that “[c]onventional systems allow *only one type of configuration* - either every node is an in-process node or every node is an out-of-process node. For example, if an out-of-process client is desired, then all other clients would also need to be configured as out-of-process clients” (Page 4, lines 20 – 23). This statement clearly and unequivocally shows that the admitted prior art does not include or suggest the specific limitations of claim 1. In fact, conventional systems are purposefully designed to allow only one type of configuration - either every node is an in-process node or every node is an out-of-process node.

In the Response to Argument section of his Answer, the Examiner reiterates those portions of the cited art, both AAPA and Jin, relied on for the Examiner's rejection. The Examiner does not comment on Appellants assertion that AAPA teaches away from the Appellants' invention and against the Examiner's particular combination of AAPA and Jin. Instead, the Examiner merely repeats his previous citations of AAPA and Jin and also repeats that it would have been obvious to modify the teachings of AAPA in view of Jin. The Examiner does not provide any explanation as to why the Examiner's combination of AAPA and Jin would be obvious in light of the fact that AAPA expressly teaches away from such a combination. It is clear that the references, whether considered alone or in combination, do not teach or suggest a distributed data system comprising a plurality of nodes including at least one in-process node and at least one out-of-process node, as recited in claim 1. Since the AAPA reference explicitly teaches away from the

claimed combination of limitations, and since the Examiner agrees that Jin does not teach the claimed limitations under discussion, the art cited by the Examiner, when considered in its totality, fails to render the claimed combination of limitations obvious. Thus, the Examiner is only considering select portions of the AAPA while ignoring the overall fact that AAPA teaches away from a combination with Jin, as relied on by the Examiner.

As noted in Appellants' appeal brief, "[r]eferences that teach away cannot service to create a *prima facie* case of obviousness." *In re Gurley*, 27 F.3d 551, 553, 31 USPQ2d 1131, 1132 (Fed. Cir. 1994). The Examiner has previously responded to this argument asserting, "a prior art reference must be considered in its entirety, including portions that would lead away from the claimed invention". However, when the AAPA reference is taken in its entirety, it clearly and unequivocally teaches away from the combination of limitations recited in claim 1, and therefore **cannot create a *prima facie* case of obviousness**. The AAPA explicitly states that that conventional systems allow only one type of configuration - either every node is an in-process node or every node is an out-of-process node. This is explicitly counter to Appellants' claims. Nor does Jin suggest a system comprising both an in-process node and an out-of-process node as recited in claim 1.

Additionally, not only does one cited art reference (AAPA) teach away from Appellants' claimed invention, the other cited art reference (Jin), whether considered alone or in combination with AAPA, also fails to teach or suggest other features of Appellants' claimed invention. For example, Jin, even if combined with AAPA, does not teach or suggest an in-process node of a distributed data system, where the in-process node comprises a client and a distributed data manager configured to execute within the same process.

In response to this argument, when presented previously, the Examiner responds by stating that the Examiner relies on AAPA to teach this limitation and to note that "one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references", citing *In re Keller*. However, Appellants are

not arguing the reference individually. In fact, the particular one of Appellants' argument to which the Examiner is referring is part of Appellants' larger argument showing that the Examiner's reliance on the cited art (both AAPA and Jin) is incorrect. For instance, Appellants argue that the combination of cited art does not teach or suggest the limitations of Appellants' claim. Appellants have then argued that the AAPA expressly teaches away from Appellants' claimed invention; therefore, the AAPA cannot be relied upon to establish a *prima facie* obviousness rejection. Appellants then argue that Jin also fails to teach or suggest the limitations of Appellants' claim. Appellants then conclude that the Examiner's combination of cited art, fails to teach or suggest the limitations of Appellants' claimed invention. Thus, Appellants' previous statements regarding a single reference where to part of a larger argument and to show that the Examiner's reliance on that reference is misplaced.

In response to Appellants' argument that it is improper to combine references where the references teach away from their combination, the Examiner again merely cites those portions of the cited art (both AAPA and Jin) relied on by the Examiner for the rejection of claim 1 and submits, without any further explanation or evidence, "that it is proper to combine references where the references do not teach away from their combination." Thus, the Examiner has simply restated his opinion that the references are combinable without actually addressing Appellants' argument. For instance, the Examiner fails to comment at all on the passage cited by Appellants to show that AAPA teaches away from the Examiner's combination.

Regardless of whether or not AAPA teaches away, Appellants have also shown that the Examiner's combination of AAPA and Jin would not result in Appellants' claimed invention, as recited in claim 1. For instance, Appellants pointed out in the Appeal Brief that there is no teaching or suggestion in the art of record that Jin's in-process and out-of-process "application manager" and "server" combinations would in any way apply to the configuration of clients and distributed data managers in the nodes of AAPA. The Examiner does not appear to address this argument in the Examiner's Answer. Appellants submit that Jin's use of the terms "in-process" and "out-of-process"

refers to the manner in which Jin's "application managers" responsible for processing client requests interact with Jin's "server processes".

Appellants have also argued that even if the teachings of Jin were to be combined with those of AAPA, the combination would only lead to "application managers" (client request processors) that may "run within the same process" or "different processes" as a server, not to a distributed data system in which at some nodes, clients run in the same process as distributed data managers, and at other nodes, clients run in separate processes from distributed data managers, as recited in claim 1. The Examiner does not address how Jin's application managers and servers apply to the distributed data manager and nodes of the AAPA. Instead, the Examiner merely relies on the fact that Jin uses the term in-process and out-of-process, **even though Jin uses those terms for completely different processes and methods than the in-process and out-of-process clients of the AAPA.**

Appellants also argue that the Examiner has not provided a proper motivation to combine AAPA and Jin. The Examiner states that "[o]ne would be motivated to do so to provide higher performance at the risk of crashing the system and afford high reliability" citing column 4, lines 55-62 of Jin. However, the Examiner's stated motivation is merely Jin's own description of the Advantages of Jin's system. Jin teaches that the Examiner's stated motivations are the benefits of executing his server applications either in-process with the server software, which "offers higher performance at the risk of crashing the system," or out-of-process with the server software, which "affords high reliability and crash protection at the expense of lower performance due to cross-process communication." Thus, Jin teaches providing higher performance at the risk of crashing the system is a characteristic of executing his server application in-process with server software and that affording high reliability and crash protection at the expense of lower performance is a characteristic of executing his server application out-of-process with the server software.

However, the characteristics and/or benefits regarding the specific execution of Jin's server software and server application do not provide any motivation to modify the AAPA's distributed data manager and clients, especially in light of the fact that the AAPA teaches away from such a modification. The characteristics and/or benefits listed by Jin and relied on by the Examiner apply to Jin's server system, but not to a distributed data management system. Thus, the Examiner has failed to provide a proper motivation for modifying AAPA in light of Jin's teachings. Instead, the Examiner's stated motivation only applies to Jin's server systems, and AAPA expressly teaches away from the combination proposed by the Examiner. One seeking to obtain the advantages noted by the Examiner would simply use Jin's system, not modify the AAPA. Moreover, as noted previously, it is improper to combine references where the references teach away from their combination. *In re Grasselli*, 218 USPQ 769, 779 (Fed. Cir. 1983). Therefore, the rejection is clearly improper.

**Claims 6 and 21:**

Appellants have argued in regards to claim 6, and contrary to the Examiner's assertion, that the combination of AAPA and Jin fails to teach or suggest wherein all data store operations performed by the distributed data manager in the in-process node store data in a non-serialized object format in a data store of the in-process node. Specifically, Appellants have argued that AAPA teaches, citing page 4, lines 15 – 18, that “in-process configuration data may be communicated between a distributed data manager and a client sharing the same process space, without the additional computation requirement for serialization/deserialization” and, citing lines 8 – 10, that “a requesting distributed data manager stores data in its data store and returns the data or a pointer to the client 201A indicating where the requested data is in the data store 221A”, but nowhere in the cited art is there a teaching or suggestion of all data store operations performed by a distributed data manager in an in-process node storing data in a non-serialized format in a data store of the in-process node, as recited in the claim.

In the Response to Arguments section of the Examiner's Answer, the Examiner cites paragraphs 11 and 12 of AAPA and asserts, "[o]ne of ordinary skill in the art will readily recognize that all data store operations performed by the distributed manager in the in-process node store data in a non-serialized format since communication between a distributed data manager and client without the additional [computational] requirement for serialization/deserialization. Thus, the Examiner is relying on the fact that AAPA states that communication between an in-process client and the distributed data manager on the same node may take place without the additional overhead of serialization/deserialization. This portion of the AAPA is referring to the fact that since communication between the in-process client and the distributed data manager on the same node does not *require* serialization and deserialization as required for communication between nodes and between an out-of-process client and a distributed data manager. However, not requiring serialization/deserialization is not the same as teaching that all data store operations performed by the distributed data manager in an in-process node storing data in a non-serialized format. Not requiring serialization/deserialization does not preclude a distributed data manager from storing data in a serialized format.

Moreover, the fact that communication between an in-process client and a distributed data manager on the same node does not require serialization and/or deserialization is irrelevant to the rejection of claim 6. The Examiner appears to be focusing on communication between an in-process client and a distributed data manager. However, claim 6 does not recite anything regarding the communication between an in-process client and a distributed data manager. Instead, claim 6 recites that all data store operations performed by the distributed data manager in the in-process node store data in a non-serialized object format in a data store of the in-process node. Thus, the Examiner's reliance on the teachings of AAPA regarding communication between an in-process client and an out-of-process client is irrelevant to the specific limitations of claim 6.

The rejection of claim 6 is therefore further unsupported by the cited art, and removal thereof is respectfully requested. Similar remarks also apply to claim 21.

**Second Ground of Rejection:**

With respect to claim 10, Appellants have argued that the Examiner's combination of cited art does not teach or suggest wherein the distributed data manager for the out-of-process node is configured to store the data in its serialized format. Specifically, Appellants argue that the Examiner's reliance on Crites is misplaced. The Examiner cites column 2, lines 36-42 and column 5, lines 19-20 of Crites. However, as shown in Appellants' Appeal Brief, the cited portion of Crites merely states, "the continuous media streams consist of sequences of digital data that are intended to be supplied serially to client computers and rendered by the client computers in a form that is useful to users of the client computers". Thus, Crites teaches sending media streams serially to client computers. Crites is silent regarding a client computer sending serialized data to a distributed data manager for storage in a serialized format. Crites provides an example of a continuous media stream as "a sequence of audio samples" that are sent to a "client computer" for "rendering as audible sound". In Crites, the "media stream" is sent to the client, whereas in claim 10, serialized data is sent from the client to a distributed data manager for storage.

In the Response to Argument section of the Examiner's Answer, the Examiner asserts that "data stream" is "a broadest reasonable interpretation of 'data in serialized format'" in view of Appellants' specification, citing page 3, lines 7-8. However, The cited portion of AAPA teaches that "serialization may include generating object data sequentially so that it may be transmitted as a data stream", but this does not imply that storing a "continuous media data stream", as taught in Crites, somehow represents data being stored in a serialized format.

Additionally, as noted above, in Crites, the "media stream" is sent *to the client*, whereas in claim 10, serialized data is sent from the client to a distributed data manager



for storage in its serialized format. In response, the Examiner cites paragraphs 6 – 7 of AAPA that also fail to teach the claimed limitation wherein the out-of-process client is configured to send serialized data to the distributed data manager for the out-of-process node to store data in its serialized format. The Examiner also asserts that the distributed data manager of the AAPA “may store data in its data store, wherein data is serialized before transmission, transmitted and received in its serialized format.” **However, the cited passages of the AAPA, even if combined with Crites, do not teach or suggest that the distributed data manager stores the data in its serialized format, as recited in Appellants claim.** Thus, AAPA in view Kin and Crites, fails to teach or suggest the limitations of claim 10.

Appellants have also argued that even if the teachings of Crites were somehow to be combined with those of Jin and AAPA, the resulting system would merely allow media streams to be served to clients from the nodes of a homogeneous distributed data system (i.e., one in which the nodes are either all “in-process nodes” or all “out-of-process nodes”), and would not render the combination of limitations of claim 10 obvious. In response, the Examiner cites the same portions of AAPA and Crites noted in the rejection of claim 10 and repeats the assertion that it would have been obvious to modify the teachings of the AAPA and Jin to store the data in its serialized format as in Crites. Thus, the Examiner merely repeats the same assertion without providing any additional explanation regarding how, since as shown above, none of the cited art actually suggests a distributed data manager receiving serialized data from a client and storing the data in its serialized format.

Additionally, the Examiner has not stated a proper motivation to modify the teachings of AAPA and Jin according to Crites. The Examiner in the Response to Arguments of the Examiner’s Answer (see, part G) states, “[o]ne would be motivated to [modify AAPA and Jin] to enable application programs to render continuous media streams of different types and from different sources without being aware of the details of the media streams.” However, this statement of motivation is actually the way in which Crites describes his own invention. Thus, one desiring to “render continuous media

streams of different types and from different sources without being aware of the details of the media streams” would simply use Crites’ invention since that is exactly the benefit provided by Crites’ invention. One would not be motivated to modify AAPA and Jin, as suggested by the Examiner, to “render continuous media streams of different types and from different sources without being aware of the details of the media streams.” Thus, the Examiner’s stated motivation does not provide any actual motivation to modify AAPA and Jin according to Crites.

## CONCLUSION

For the foregoing reasons submitted in the Appeal Brief and this Reply Brief, it is submitted that the Examiner's rejections of claims 1-33 is erroneous, and reversal of his decision is respectfully requested.

The Commissioner is authorized to charge any fees that may be due to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-05700/RCK. This Reply Brief is submitted with a return receipt postcard.

Respectfully submitted,

/Robert C. Kowert/

Robert C. Kowert, Reg. #39,255  
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.  
P.O. Box 398  
Austin, TX 78767-0398  
Phone: (512) 853-8850

Date: February 6, 2007